Question 1 (Q1: 50p)

In this problem, Expected Risk Minimization (ERM) classifier method is conducted in both theoretical and experimental aspects.

A 2-dimensional real-valued random vector **X** is defined as the following PDF:

$$p(\boldsymbol{x}) = P(L=0)p(\boldsymbol{x}|L=0) + P(L=1)p(\boldsymbol{x}|L=1)$$

in which, the class-conditional pdfs are:

$$p(\boldsymbol{x}|L=0) = w_1 g(\boldsymbol{x}|\boldsymbol{m}_{01}, \boldsymbol{C}_{01}) + w_2 g(\boldsymbol{x}|\boldsymbol{m}_{02}, \boldsymbol{C}_{02})$$

and

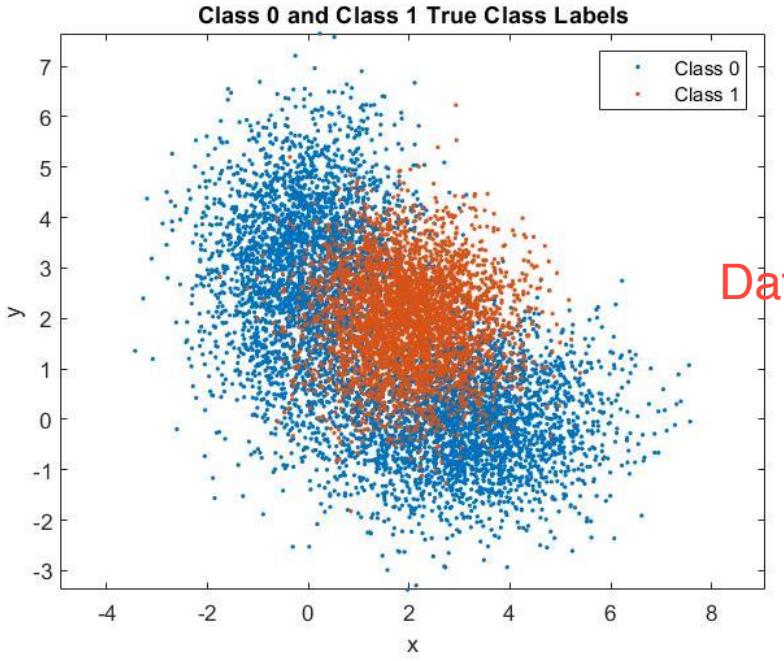
$$p(\boldsymbol{x}|L=1) = g(\boldsymbol{x}|\boldsymbol{m}_1, \boldsymbol{C}_1)$$

where g is a multivariate Gaussian probability density function with mean vector **m** and covariance matrix **C**, with L indicating the true class label and x being the sampled data. The parameters of these pdfs are:

$$P(L=0) = 0.65 \sim \begin{cases} \boldsymbol{m}_{01} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \boldsymbol{C}_{01} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} w_1 = 0.5 \\ \boldsymbol{m}_{02} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \boldsymbol{C}_{02} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} w_2 = 0.5 \end{cases}$$

$$P(L=1) = 0.35 \sim \boldsymbol{m}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Generated data samples are show in Figure 1 below:



Data: 4p

Figure 1: **X** with true labels for Question1

Part A: ERM classification using the knowledge of true data pdf (Q1a: 23p)

1. Specify the minimum expected risk classification rule in the form of a likelihood-ratio test:

$$\frac{p(\boldsymbol{x}|L=1)}{p(\boldsymbol{x}|L=0)} \underset{D=0}{\overset{D=1}{\lessgtr}} \frac{\lambda_{10}-\lambda_{00}}{\lambda_{01}-\lambda_{11}} \cdot \frac{P(L=0)}{P(L=1)} = \gamma$$

where the assumption that "0-1 loss" matrix is conducted. So, $\lambda_{10} = \lambda_{01} = 1$ and $\lambda_{00} = \lambda_{11} = 0$. Thus, the result of this likelihood-ratio is:

$$\frac{p(\boldsymbol{x}|L=1)}{p(\boldsymbol{x}|L=0)} \underset{D=0}{\overset{D=1}{\lessgtr}} \frac{1}{1} \cdot \frac{0.65}{0.35} = 1.8571 = \gamma$$

3p

2. Plot the ROC curve of the minimum expected classifier:

The classifier is applied to plot the ROC curve of expected risk minimization, which is shown in Figure 2. In this figure, the estimated (calculated) minimum error and the theoretical minimum error are marked on the curve.

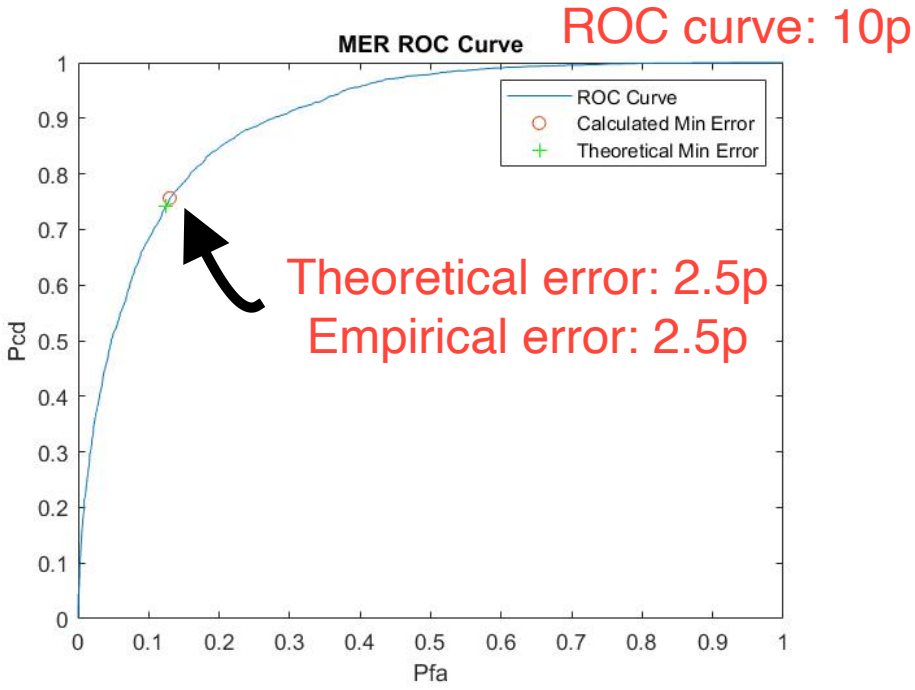


Figure 2: Expected Risk Minimization ROC curve for Question1 with the known data

3. Comparison between theoretical and estimated results in ERM:

Based on 10K data samples, the minimum probability of error that estimated is 0.1666 at the threshold γ is 1.8487.

The theoretical probability of error is calculated as the following

$$P_{error} = P_{fa} \cdot P(L = 0) + (1 - P_{cd}) \cdot P(L = 1) = 0.1669$$

where P_{fa} is the probability of false alarm, and P_{cd} is the probability of correct detection, at the threshold $\gamma = 1.8571$.

Table 1 shows the comparison between theoretical and estimated results. In Figure 2, theoretical minimum probability of error is marked as green plus sign, while the estimated one as red circle.

Table 1: Comparison between theoretical and estimated results of ERM

	γ	Min. P_{error}
Theoretical	1.8571	0.1669 2.5
Estimated	1.8487	0.1666 2.5

Part B: Fisher Linear Discriminant Analysis (Q1b: 23p)

In Fisher LDA, the projection weight vector $\boldsymbol{w}_{LDA}$ maximizes the ratio of the difference between the mean and covariance of sampled data. Thus, through LDA, the dimension of the original data is reduced and the classification problem is simplified. The Fisher LDA rule is shown as the following:

$$\boldsymbol{w}_{LDA} \cdot \boldsymbol{x} \underset{D=0}{\overset{D=1}{\lessgtr}} \tau$$

where $\boldsymbol{w}_{LDA}$ is defined as the following:

$$\boldsymbol{S}_w \cdot \boldsymbol{w}_{LDA} = \frac{\boldsymbol{w}_{LDA}^T \cdot \boldsymbol{S}_w \cdot \boldsymbol{w}_{LDA}}{\boldsymbol{w}_{LDA}^T \cdot \boldsymbol{S}_B \cdot \boldsymbol{w}_{LDA}} \cdot \boldsymbol{S}_B \cdot \boldsymbol{w}_{LDA}$$

where

$$\boldsymbol{S}_B = (\boldsymbol{m}_0 - \boldsymbol{m}_1)(\boldsymbol{m}_0 + \boldsymbol{m}_1)^T$$
$$\boldsymbol{S}_w = \boldsymbol{C}_0 + \boldsymbol{C}_1$$

Based on the rule, the projected data samples from different given classes of pdfs are shown in Figure 3. In order to enhance the visibility of data, Figure 4 is shown to separately plot the projection data from two classes. Besides, the threshold corresponding to the minimum probability of error is marked as dash line in Figure 4.

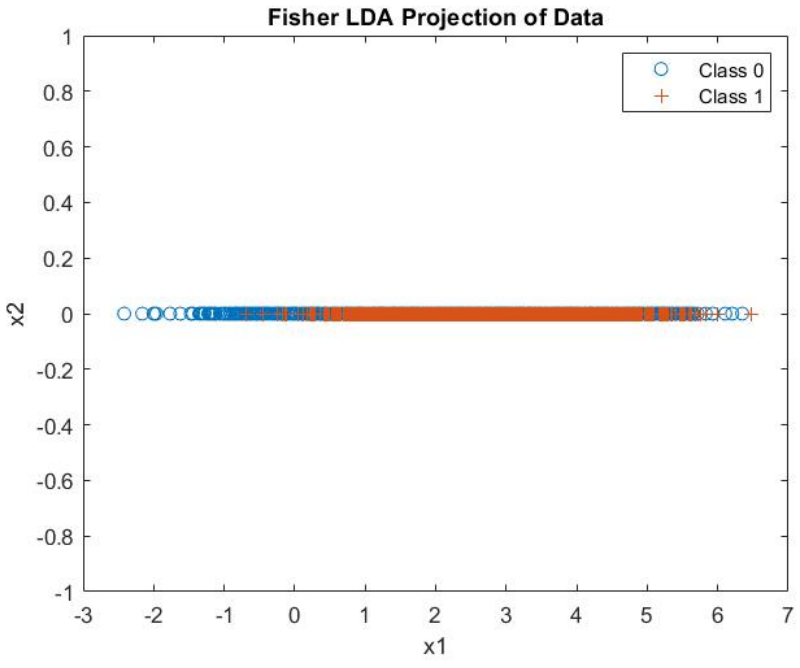


Figure 3: Fisher LDA projection of data from given classes

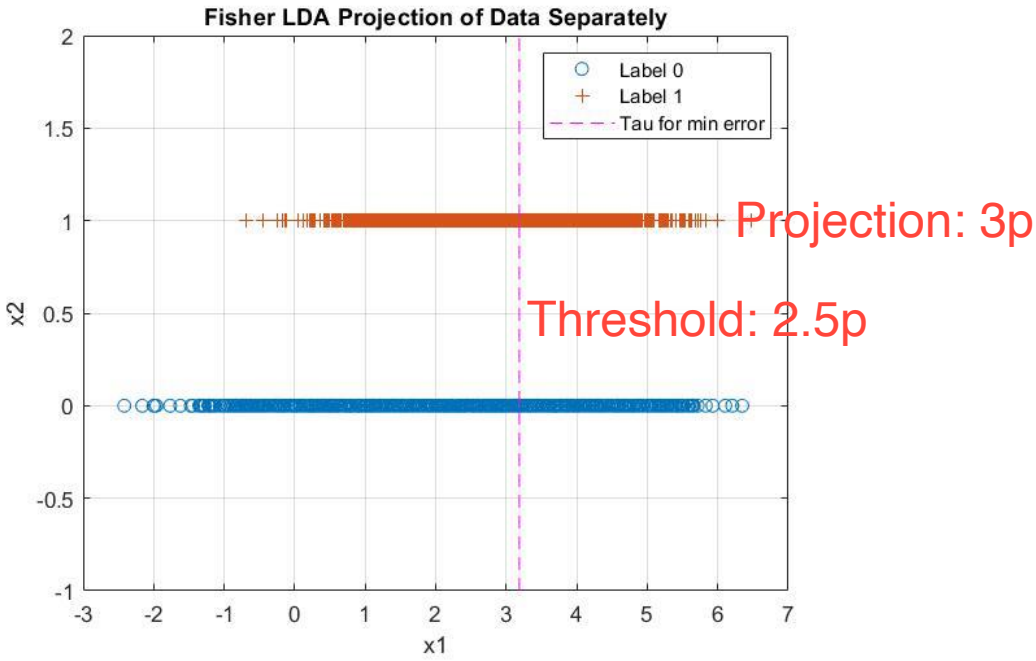


Figure 4: Separate Fisher LDA projection of data samples with minimum probability of error corresponding threshold

The LDA classification rule is applied to plot the ROC curve, which takes values from $-\infty\ to\ +\infty$, shown in Figure 5. In this figure, the estimated (calculated) minimum probability of error is marked as a red circle.

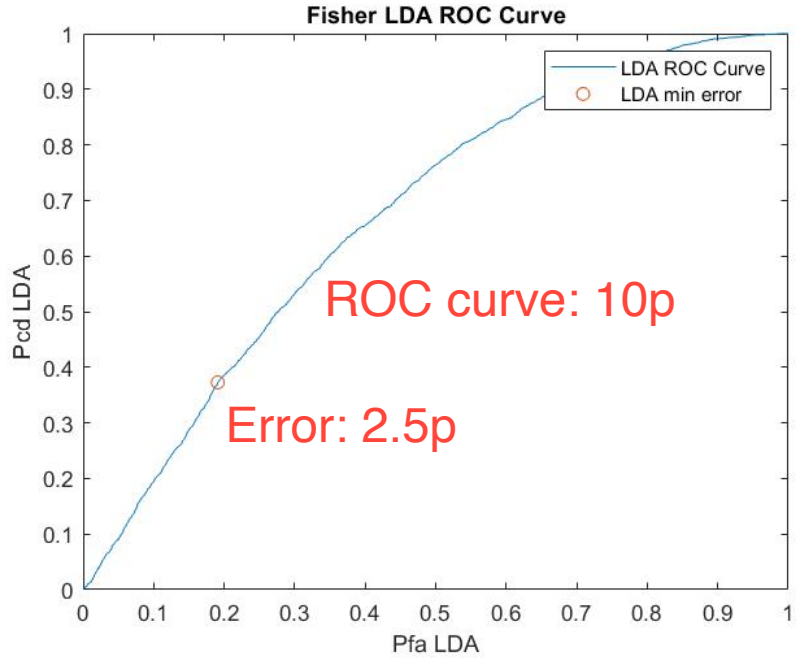


Figure 5: Fisher LDA ROC curve for Question1 with the known data

Based on 10K data samples in LDA problem, the minimum probability of error that estimated is 0.3438 at the threshold γ is 3.3469.

If in the best scenario, all data samples from true label 1 would be classified wrong. In this case, the probability of false alarm will be 0 and the probability of correct detection will also be 0. Thus, the theoretical probability of error is calculated as the following

$$P_{error} = P_{fa} \cdot P(L=0) + (1 - P_{cd}) \cdot P(L=1) = 0 \cdot 0.65 + 1 \cdot 0.35 = 0.35$$

Table 2 shows the comparison between theoretical and estimated results in Fisher LDA case.

Table 2: Comparison between theoretical and estimated results of Fisher LDA

	τ	Min. P_{error}
Theoretical	1.8571	0.35
Estimated	3.3469	0.3438

2.5

2.5

Question 2 (Q2: 50p)

In this problem, minimum probability of error classification (EM) and ERM are conducted. Different loss matrices are applied to evaluate confusion matrix and show the effect as a result.

Part A: Minimum probability of error classification (0-1 loss) (Q2a: 30p)

1. Parameters design and data generation:

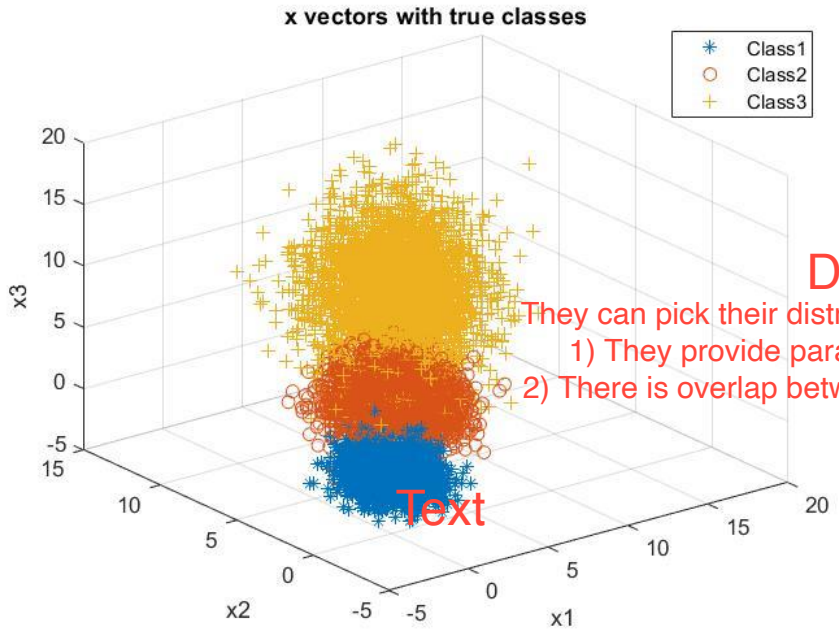
A 3-dimensional real-valued random vector **X** is defined as a mixture of 4 Gaussian distribution, where class 3 data is generated from a mixture of Gaussian distribution 3 and 4. The parameters (mean, covariance, priors and weight) of vector **X** is designed as the following:

$$P(L=1) = 0.3 \sim \boldsymbol{m}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{C}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P(L=2) = 0.3 \sim \boldsymbol{m}_2 = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} \boldsymbol{C}_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$P(L=3)=0.4 \sim \begin{cases} \boldsymbol{m}_3=\begin{bmatrix}6\\6\\6\end{bmatrix} & \boldsymbol{C}_3=\begin{bmatrix}3&0&0\\0&3&0\\0&0&3\end{bmatrix} & w_3=0.5 \\ \boldsymbol{m}_4=\begin{bmatrix}9\\9\\9\end{bmatrix} & \boldsymbol{C}_4=\begin{bmatrix}4&0&0\\0&4&0\\0&0&4\end{bmatrix} & w_4=0.5 \end{cases}$$

where 4 different Gaussian probability density function with mean vector **m** and covariance matrix **C**, with L indicating the true class label and x being the sampled data. Generated data samples are show in Figure 6 below:



Data: 5p
They can pick their distributions, give full score if
1) They provide params (top of this page)
2) There is overlap between classes (this graph)

Figure 6: True data distributions for generated samples of Question2

2. Specify the decision rule of minimum probability of error:

$$D(x)=\underset{d\in\{1,2,3\}}{\operatorname{argmin}}\sum_{l=1}^{C}\lambda_{dl}p(\boldsymbol{x}|L=l)P(L=l)$$

Decision rule: 5p

where C is the number of classes, which is 3 in this problem. λ indicates the loss will incur by deciding d given samples from class l. $p(\boldsymbol{x}|L=l)$ is the class condition, which is the likelihood of data sample given it from class l, and $P(L=l)$ is the class prior.

To demonstrate this decision rule, the goal is to choose the decision with the minimum $R(D=d|x)$ as the best solution. Thus, the vector form is described as the following:

$$R(D=1\dots A|x)=\Lambda P(L=1\dots C|x)$$

where R indicates all decision risks across option, Λ is loss matrix and P indicates all class posteriors that computed from conditional pdfs and class priors.

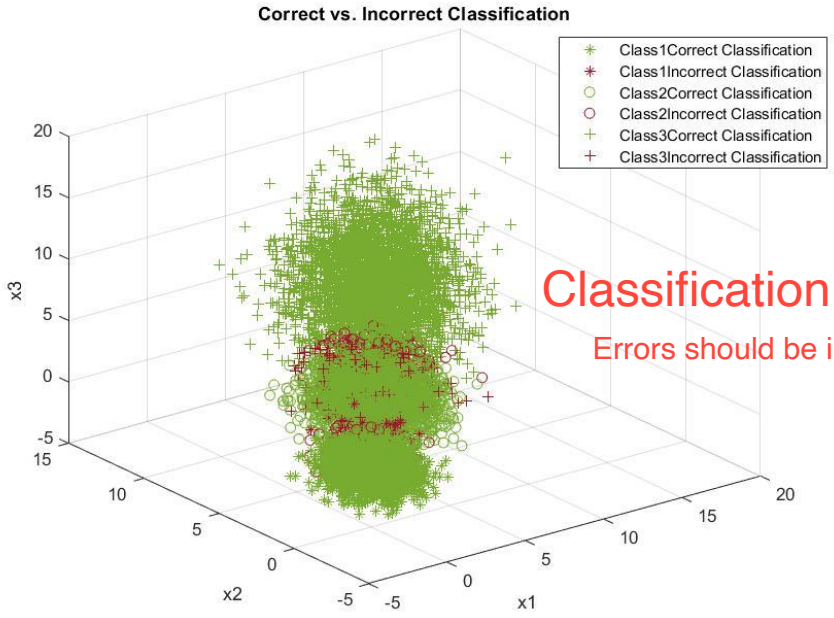
3. Provide a visualization of data with 0-1 loss matrix:

The loss matrix for a 0-1 loss problem is chosen as the ideal matrix to minimize the probability of error, which is:

$$\Lambda = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The diagonal of this loss matrix shows correct classification will cost no loss since decision=class label. While the rest of the matrix elements shows the cost of wrong classification is 1 corresponding to decision≠class label.

The data and the result of classification with a 0-1 loss matrix is shown in Figure 7. As for 3 different classes, the markers of these data distribution are used as: class1-star sign, class2-circle and class3-plus sign. Besides, green markers indicates the correct classification and red shows the wrong ones.



Classification error: 10p

Errors should be in between classes.

Figure 7: Data distribution with the result of classification with 0-1 loss matrix

Confusion matrix shows the result for the minimum error loss with the probability, which is shown below:

Large numbers on the diagonal, small otehrwise.
Should be consistent with classification error plot (fig7).
For each column: sum(column)=1

$$\text{Confusion Matrix(i, j)} = \begin{bmatrix} 0.9878 & 0.0175 & 0.0005 \\ 0.0122 & 0.9580 & 0.0489 \\ 0 & 0.0245 & 0.9506 \end{bmatrix}$$

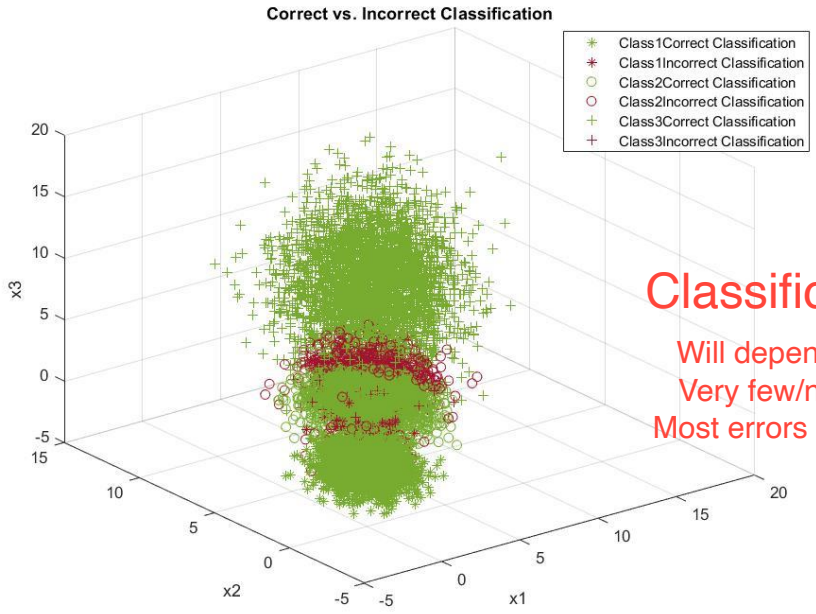
Confusion matrix: 10p

in which, the row index i indicates the decision and column index j indicates the true class label.

Part B: ERM classification with different loss matrices corresponding to Label 3
The loss matrix (Q2b: 20p)

$$\Lambda_{10} = \begin{bmatrix} 0 & 1 & 10 \\ 1 & 0 & 10 \\ 1 & 1 & 0 \end{bmatrix}$$

is used to repeat the previous procedure and to see the changes regarding to the new value. The new result of data classification is shown in Figure 8, followed with the confusion matrix.



Classification:5p
Will depend on their data but:
Very few/no errors on class 3
Most errors should be class 2->3

Figure 8: Data distribution with the result of classification with loss matrix Λ_{10}

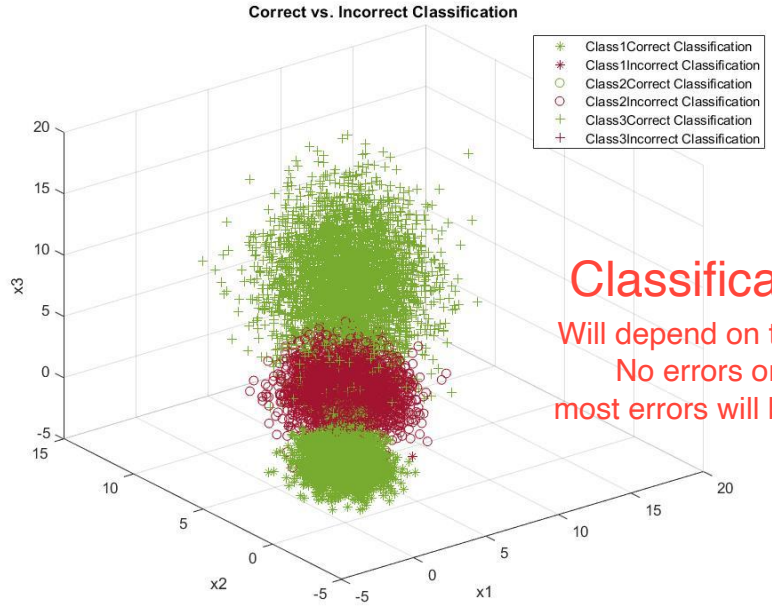
$$\text{Confusion Matrix}_{10}(i,j) = \begin{bmatrix} 0.9878 & 0.0175 & 0.0005 \\ 0.0122 & 0.7944 & 0.0195 \\ 0 & 0.1880 & 0.9800 \end{bmatrix}$$

Decision matrix: 5p
Will depend on their data, but large number C33 and class 2 classified as 3 (C22) smaller than PART A, C23 bigger than PART A.
Each colum:sum(column)=1

Then, the loss matrix

$$\Lambda_{100} = \begin{bmatrix} 0 & 1 & 100 \\ 1 & 0 & 100 \\ 1 & 1 & 0 \end{bmatrix}$$

is used to repeat the previous procedure. New result of data classification is shown in Figure 9, followed with the confusion matrix.



Classification: 5p

Will depend on their data but:
No errors on class 3,
most errors will be class 2 ->3

Figure 9: Data distribution with the result of classification with loss matrix Λ_{100}

Decision matrix: 5p

$$\text{Confusion Matrix}_{100}(i,j) = \begin{bmatrix} 0.9871 & 0.0149 & 0.0005 \\ 0.0030 & 0.0684 & 0.0003 \\ 0.0099 & 0.9167 & 0.9992 \end{bmatrix}$$

Will depend on their data, but C33 larger than part B, (C22) smaller than PART B, C23 bigger than PART B
Sum(column)=1

Regarding to the changes of the loss matrix from Λ to Λ_{10} and to Λ_{100}, an increasing cost for misclassification when class label = 3 is conducted. Therefore, in order to minimize the overall probability of error, the precision and accuracy of label 3 classification has to be improved to ensure that as many data distributions as possible for class 3 are correctly classified.

So, in the confusion matrix corresponding to these three different loss matrices, the probability of label 3 correctly classified is increasing as the cost for wrong classified is increasing.

Appendix: Matlab code for Homework 1

```
%% Question 1 Setup %%

clear all;
close all;
clc;

n = 2;     %Dimensions of data
N = 10000;  %Number of data samples
p0 = 0.65;  %Prior for label 0
p1 = 0.35;  %Prior for label 1
w1 = 0.5;   %Weight for label 0 condition 1
w2 = 0.5;   %Weight for label 0 condition 2

u = rand(1,N)>=p0;  %Determine posteriors

%Create appropriate number of data points from each
distribution
N0 = length(find(u==0));
N1 = length(find(u==1));
N = N0+N1;
label = [zeros(1,N0) ones(1,N1)];

%Parameters for two classes
m01 = [3;0]; C01 = [2,0;0,1];
m02 = [0;3]; C02 = [1,0;0,2];
m1 = [2;2]; C1 = [1,0;0,1];

%Data generation with requirement
gmmParameters.priors = [w1,w2];
gmmParameters.meanVectors = [m01,m02];
gmmParameters.covMatrices(:,:,1) = C01;
gmmParameters.covMatrices(:,:,2) = C02;
[r0,P0_label] = generateDataFromGMM(N0,gmmParameters);
r1 = mvnrnd(m1,C1,N1)';

%Combine data into a single dataset
x = zeros(n,N);
x(:,label==0) = r0;
x(:,label==1) = r1;

%Plot data showing two classes
```

```
figure;
plot(r0(1,:),r0(2,:),'.');
axis equal;
hold on;
plot(r1(1,:),r1(2,:),'.');
title('Class 0 and Class 1 True Class Labels')
xlabel('x')
ylabel('y')
legend('Class 0','Class 1')

%% Question 1 Part A %%

%Calculate discriminant scores and tau
disScore =
log(evalGaussian(x,m1,C1)./(w1*evalGaussian(x,m01,C01)+w2
*evalGaussian(x,m02,C02)));
%gamma = sort(disScore(disScore>=0));
%tau = log(gamma);

%Generate vector of threshold for parametric sweep
tau = [min(disScore)-eps sort(disScore)+eps];

%Make decision for every threshold and calculate error
values
for i = 1:length(tau)
    decision = disScore >= tau(i);
    Pfa(i) = sum(decision==1 & label==0)/N0;
    Pcd(i) = sum(decision==1 & label==1)/N1;
    P_error(i) = Pfa(i)*p0+(1-Pcd(i))*p1;
end

%Find minimum error and corresponding threshold
[min_error,min_index] = min(P_error);
min_decision = (disScore >= tau(min_index));
min_fa = Pfa(min_index);
min_cd = Pcd(min_index);

%Find theoretical minimum error(threshold calculated
using class priors)
theo_decision = disScore>=log(p0/p1);
theo_Pfa = sum(theo_decision==1 & label==0)/N0;
theo_Pcd = sum(theo_decision==1 & label==1)/N1;
theo_error = theo_Pfa*p0+(1-theo_Pcd)*p1;
```

```
%Plot ROC curve with min error point labeled
figure(2);
plot(Pfa,Pcd,'-
',min_fa,min_cd,'o',theo_Pfa,theo_Pcd,'g+');
title('MER ROC Curve');
legend('ROC Curve','Calculated Min Error','Theoretical
Min Error');
xlabel('Pfa');
ylabel('Pcd');

%% Question 1 Part B %%
% Estimate mean vectors and covariance matrices from
samples
mu0hat = mean(r0,2); S0hat = cov(r0');
mu1hat = mean(r1,2); S1hat = cov(r1');

% Calculate the between/within-class scatter matrices
Sb = (mu0hat-mu1hat)*(mu0hat-mu1hat)';
Sw = S0hat + S1hat;

% Solve for the Fisher LDA projection vector (in w)
[V,D] = eig(inv(Sw)*Sb);
[~,ind] = sort(diag(D),'descend');
w = V(:,ind(1));    % Fisher LDA projection vector
y = w'*x;          %All data projected on to the line
spanned by w

w = sign(mean(y(label==1))-mean(y(label==0)))*w;
y = sign(mean(y(label==1))-mean(y(label==0)))*y;

%Evaluate for different taus_LDA
tau_LDA = [min(y)-eps sort(y)+eps];

for j = 1:length(tau_LDA)
    decision_LDA = y>tau_LDA(j);
    Pfa_LDA(j) = sum(decision_LDA==1 & label==0)/N0;
    Pcd_LDA(j) = sum(decision_LDA==1 & label==1)/N1;
    P_error_LDA(j) = Pfa_LDA(j)*p0+(1-Pcd_LDA(j))*p1;
end

%Find min error and corrspoding threshold
[min_error_LDA, min_index_LDA] = min(P_error_LDA);
min_decision_LDA = (y >= tau_LDA(min_index_LDA));
min_Pfa_LDA = Pfa_LDA(min_index_LDA);
```

```
min_Pcd_LDA = Pcd_LDA(min_index_LDA);

% Plot LDA projections and LDA ROC curve
figure(3);
plot(y(label==0),zeros(1,N0),'o',y(label==1),zeros(1,N1),
'+');
title('Fisher LDA Projection of Data');
xlabel('x1');
ylabel('x2');
legend('Class 0','Class 1');
hold on;

figure(4);
plot(Pfa_LDA,Pcd_LDA,'-',min_Pfa_LDA,min_Pcd_LDA,'o');
title('Fisher LDA ROC Curve');
legend('LDA ROC Curve','LDA min error');
xlabel('Pfa LDA');
ylabel('Pcd LDA');

figure(5);
plot(y(label==0),zeros(1,N0),'o','DisplayName','Label
0');
hold all;
plot(y(label==1),ones(1,N1),'+','DisplayName','Label 1');
ylim([-1 2]);
plot(repmat(tau_LDA(min_index_LDA),1,2),ylim,'m--
','DisplayName','Tau for min error');
grid on;
xlabel('x1');
ylabel('x2');
title('Fisher LDA Projection of Data Separately');
legend 'show';
```

```
%% Question 2 Setup %%

clear all;
close all;
clc;

C = 3;
N = 10000;
n = 3;

%Parameters for each distribution
priors = [0.3 0.3 0.4];
weight = [0.5 0.5];

mu(:,1) = [0;0;0];
mu(:,2) = [3;3;3];
mu(:,3) = [6;6;6];
mu(:,4) = [9;9;9];

S(:,:,1) = eye(3);
S(:,:,2) = 2*eye(3);
S(:,:,3) = 3*eye(3);
S(:,:,4) = 4*eye(3);

%Define loss matrix
%lossMatrix = {'minErr' 'lambda_10' 'lambda_100'};
lossMatrix(:,:,1) = ones(C,C)-eye(C);
lossMatrix(:,:,2) = [0 1 10; 1 0 10; 1 1 0];
lossMatrix(:,:,3) = [0 1 100; 1 0 100; 1 1 0];

%% Question 2 Part A %%
%Create appropriate number of data points from each
distribution(label)
label = rand(1,N);
for ind = 1:length(label)
    if label(ind) < priors(1)
        label(ind) = 1;
    elseif label(ind) < (priors(1)+priors(2))
        label(ind) = 2;
    elseif label(ind) <
(priors(1)+priors(2)+weight(1)*priors(3))
        label(ind) = 3;
    else
        label(ind) = 4;
```

```
    end
end
N1 = length(find(label==1));
N2 = length(find(label==2));
N3 = length(find(label==3))+length(find(label==4));

%Data generation with requirement
C3_gmmP.priors = weight;
C3_gmmP.meanVectors = mu(:,3:4);
C3_gmmP.covMatrices = S(:,:,3:4);
[x3,~] = generateDataFromGMM(N3, C3_gmmP);
%x1 = mvnrnd(mu(:,1),S(:,:,1),N1)';
%x2 = mvnrnd(mu(:,2),S(:,:,2),N2)';

m(:,3) = mean(x3,2);
Sigma(:,:,3) = cov(x3');
m(:,1:2) = mu(:,1:2);
Sigma(:,:,1:2) = S(:,:,1:2);

gmmP.priors = priors;
gmmP.meanVectors = m;
gmmP.covMatrices = Sigma;
[x,truelabel] = generateDataFromGMM(N,gmmP);
for ind = 1:C
    Nclass(ind,1) = length(find(truelabel==ind));
end

%Shared computation for both parts
for ind = 1:C
    pxgivenl(ind,:) =
evalGaussianPDF(x,gmmP.meanVectors(:,ind),gmmP.covMatrice
s(:,:,ind));
end

px = gmmP.priors*pxgivenl; % Total probability theorem
classPosteriors =
pxgivenl.*repmat(gmmP.priors',1,N)./repmat(px,C,1); %P(L=
1|x)

%Plot data with true labels
sym = ['*','o','+'];
%col = ['g','r','r';'r','g','r';'r','r','g'];

figure(1);
```

```
for ind = 1:C
    plot3(x(1,truelabel==ind),x(2,truelabel==ind),...
        x(3,truelabel==ind),sym(ind),'DisplayName',...
        ['Class' num2str(ind)]);
    hold on;
end
xlabel('x1');
ylabel('x2');
zlabel('x3');
grid on;
title('x vectors with true classes');
legend 'show';


%% Question 2 Part B %%

%Classify data based on loss matrix
for ind = 1:3

ER = lossMatrix(:,:,ind)*classPosteriors;
[~,decision] = min(ER,[],1);

figure
for i = 1:C %each decision
    for j = 1:C %each class label
        cfMatrix(i,j,ind) = sum(decision==i &
truelabel==j)/Nclass(j);
        if i == j
            p(i,j)=scatter3(x(1,decision==i &
truelabel==j),x(2,decision==i & truelabel==j),...
                x(3,decision==i &
truelabel==j),sym(j),'MarkerEdgeColor','#77AC30','Display
Name',...
                ['Class' num2str(j) 'Correct
Classification']);
            hold on
        else
            p(i,j)=scatter3(x(1,decision==i &
truelabel==j),x(2,decision==i & truelabel==j),...
                x(3,decision==i &
truelabel==j),sym(j),'MarkerEdgeColor','#A2142F','Display
Name',...
                ['Class' num2str(j) 'Incorrect
Classification']);
```

```
            hold on;
        end
    end
end
title('Correct vs. Incorrect Classification');
legend ([p(1) p(2) p(5) p(4) p(9) p(7)]);
grid on;
xlabel('x1');
ylabel('x2');
zlabel('x3');

end
```